The 36th Annual ACM

International Collegiate Programming Contest

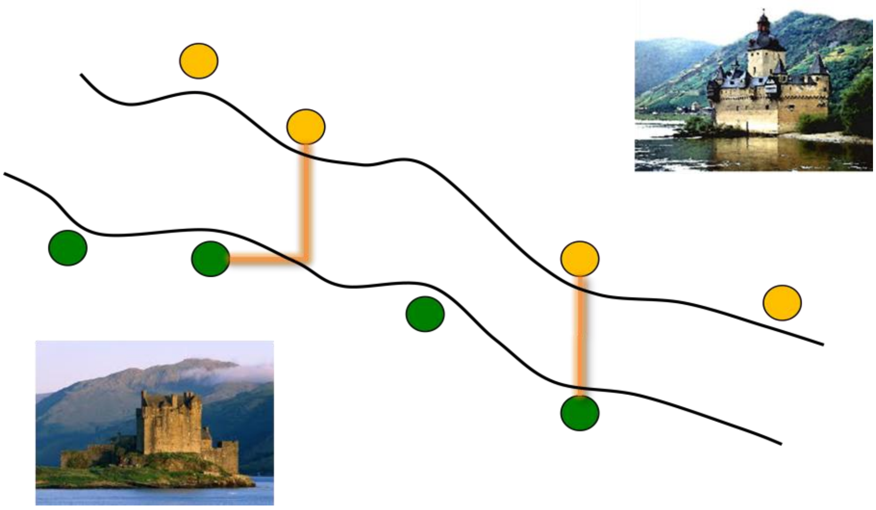
Asia Regional – Daejeon

Nationwide Internet Competition

Problem A

Castles

The country ICPCIA has a river called “river of castles”. In the past, ICPCIA was divided into two kingdoms called Westeria and Eastania. Two kingdoms were separated by the river which runs in the direction from northwest to southeast. Both kingdoms had competitively built many castles for defense and attack along the riverside. All castles of each side of the river are located with an x-monotone increasing and y-monotone decreasing feature. In other words, let $S = \{s_1, s_2, \ldots, s_n\}$ be the castles of Westeria and $T = \{t_1, t_2, \ldots, t_m\}$ the castles of Eastania. Let (x_i, y_i) (resp. (u_i, v_i)) be the coordinates of s_i (resp. t_i). Then $x_i < x_j$ and $y_i > y_j$ if $i < j$. Also, $u_i < u_j$ and $v_i > v_j$ if $i < j$.



Now, the Ministry of Culture and Tourism of ICPCIA decided to build a beautiful bridge connecting two castles on the opposite side of the river. The bridge will be an I-shape(a horizontal segment or a vertical segment) or an L-shape(both of a horizontal segment and a vertical segment). They want to find the location of the bridge such that its length is as small as possible. So, they try to find the closest pair of castles on the opposite side. The distance between two castles s_i and t_j is computed by $|x_i - u_j| + |y_i - v_j|$.

Given the information of two sets of castles, write a program to find the distance between the closest castles on the opposite side of the river.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case consists of three lines. The first line of each test case contains two integers. The first integer, n, is the number of castles in the west side of the river, and the integer, m, is the number of castles in the east side of it, where $1 \le n, m \le 200{,}000$. The second line of each test case contains $2n$ integers $x_1, y_1, x_2, y_2, \ldots, x_n, y_n$, where (x_i, y_i) is the coordinate of the i-th castle in the west side of the river and $x_i < x_j$ and $y_i > y_j$ if $i < j$. The last line of each test case contains $2m$ integers $u_1, v_1, u_2, v_2, \ldots, u_m, v_m$, where (u_i, v_i) is the coordinate of the i-th castle in the east side of the river and $u_i < u_j$ and $v_i > v_j$ if $i < j$. You may assume that there exists an *x*-monotone increasing and *y*-monotone decreasing path which separates two sets of castles. All integers representing the coordinates of castles are between -10^9 and 10^9, inclusive.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain the distance between the closest castles on the opposite side of the river.

The following shows sample input and output for two test cases.

Sample Input	Output for the Sample Input
2 4 4 1 6 3 4 5 3 6 1 2 7 6 6 7 4 8 3 5 4 1 11 3 8 5 7 6 4 9 3 3 10 6 8 7 4 10 1	2 1

Problem B
Cross Country

Cross country running is a sport in which runners race on open-air courses over natural terrain. The course, typically 4–12 kilometers long, may include surfaces of grass and earth, pass through woodlands and open country, and include hills and flat ground. It is both an individual and a team sport, as runners are judged on an individual basis and a points scoring method for teams.

Scores are determined by summing the top four individual finishing places on each team and a team consists of six runners. Points are awarded to the individual runners of eligible teams, equal to the position in which they cross the finish line (first place gets 1 point, second place gets 2 points, etc.). The points for these runners are summed, and the low score wins. Individual athletes and athletes from incomplete teams are excluded from scoring. Ties are resolved in favor of the team whose 5^{th} scoring member finishes nearer to first place.

For example, the following is a table of individual results.

Place	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Team	A	B	C	C	A	C	B	D	A	A	C	A	C	C	A
Score	1	n/a	2	3	4	5	n/a	n/a	6	7	8	9	10	11	12

Teams B and D are incomplete, so not eligible for scoring. The score of team A is 18 (1+4+6+7) and the score of team C is 18 (2+3+5+8). In this case, team A and C are tie. Due to the score of 5^{th} runner of team A is less than the score of 5^{th} runner of team C, team A is the winning team.

Given the places of all runners, write a program that finds the winning team. Note that if the number of runners for each team is less than six, the team would be excluded from scoring. We assume that there is no team whose runners are more than six and at least one team has six runners and all participants finish the race.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case consists of two lines. The first line of each test case contains an integer N ($6 \le N \le 1{,}000$) which represents the number of runners. The second line of each test case contains N integers $t_1, t_2, \ldots, t_N$, which represent the team number. Each team is represented by an integer from 1 to M ($1 \le M \le 200$).

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain the winning team number.

The following shows sample input and output for two test cases.

Sample Input	**Output for the Sample Input**
2 15 1 2 3 3 1 3 2 4 1 1 3 1 3 3 1 18 1 2 3 1 2 3 1 2 3 3 3 3 2 2 2 1 1 1	1 3

Problem C
Cube of a Graph

The *cube* G^3 of a graph $G = (V, E)$ is the graph on the vertex set V in which two vertices are joined by an edge if their distance in G is at most three, where the distance between two vertices in a graph is defined as the number of edges in a shortest path connecting them. A *bridge* (also known as a cut-edge) of a graph is an edge whose deletion increases the number of connected components. Equivalently, an edge is a bridge if and only if it is not contained in any cycle.

To study hamiltonian properties of the cubes of connected graphs, the following notions concerned with bridge were suggested in early 1960's. A bridge of a graph G is said to be *nontrivial* if neither vertex incident with the edge is of degree one, where the degree of a vertex is the number of edges incident to it. See Figure 1.

- A vertex of G is called a *pure bridge vertex* if each edge incident to the vertex is a nontrivial bridge.
- A set of three distinct mutually adjacent vertices, each of degree at least three, is called a *pure bridge triangle* if each edge of G that is incident with exactly one of three vertices is a nontrivial bridge.

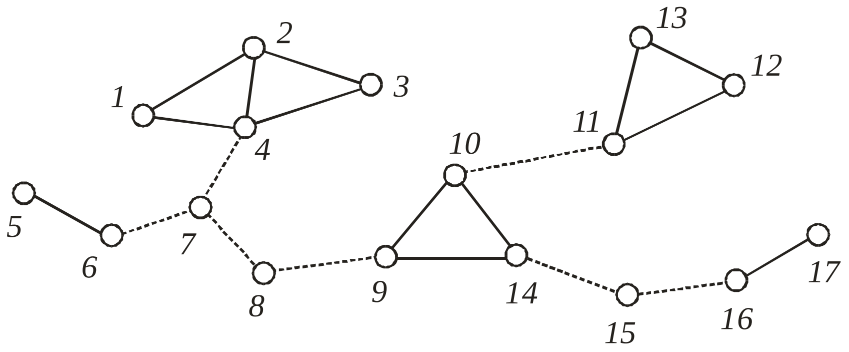


Figure 1. A connected graph which has seven nontrivial bridges represented by dotted lines. There are three pure bridge vertices 7, 8, and 15, one pure bridge triangle $\{9, 10, 14\}$, and one pure bridge pair $\{7, 8\}$.

It has been known that the cube of any connected graph G is hamiltonian-connected, i.e. every two vertices of G^3 are connected by a hamiltonian path. Recently, an ambitious research project on strong hamiltonian properties of graphs was initiated by Prof. Cho, who is a highly considered graph theorist. He introduced a new notion called pure bridge pair to characterize some interesting hamiltonian properties.

- A set of two adjacent vertices is called a *pure bridge pair* if both vertices are pure bridge vertices.

To support Prof. Cho's research, you are to write an efficient running program to identify all the pure bridge vertices, pure bridge triangles, and pure bridge pairs in a large connected graph.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. The first line of each test case contains two positive integers. The first integer n is the number of vertices and the second integer m is the number of edges in an input graph G, where $n \leq 3{,}000$ and $m \leq 1{,}000{,}000$. In the following m lines, each line contains two integers u and v which means (u, v) is an edge of G. You may assume that G is connected and the vertex set of G is $\{1, 2, \ldots, n\}$.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain the numbers of pure bridge vertices, pure bridge triangles, and pure bridge pairs in sequence.

The following shows sample input and output for two test cases.

Sample Input	Output for the Sample Input
2 5 4 2 3 1 2 3 4 5 4 17 20 1 2 2 3 3 4 4 1 2 4 4 7 5 6 6 7 7 8 8 9 9 10 10 14 14 9 10 11 11 12 12 13 13 11 14 15 15 16 16 17	1 0 0 3 1 1

Problem D
DSLR

Here is a simple calculator providing only four commands, D, S, L, and R. This calculator has only one register for storing a non-negative integer of four decimal digits. Each command modifies the integer n stored in the register as follows. Let us assume the four decimal digits of n from left to right are d_1, d_2, d_3, and d_4 (i.e. $n = ((d_1 \times 10 + d_2) \times 10 + d_3) \times 10 + d_4$).

(1) D: D doubles up n. If the result is greater than 9999, modulo 10000 is taken. As a result, the new value ($2n$ mod 10000) is stored to the register.
(2) S: S subtracts one from n and stores the result $n - 1$ to the register. If n is zero, 9999 is stored to the register instead.
(3) L: L rotates each of four digits of n to the left and stores the result to the register. After this operation, the four digits of the value of the register from left to right are d_2, d_3, d_4, and d_1.
(4) R: R rotates each of four digits of n to the right and stores the result to the register. After this operation, the four digits of the value of the register from left to right are d_4, d_1, d_2, and d_3.

As noted above, L and R commands assume the decimal representation. For example, if n = 1234, applying L results in 2341 and applying R results in 4123 to be stored in the register.

Given two different integers A and B ($A \neq B$), you are to write a program to generate a minimal sequence of commands changing the register contents from A to B. For instance, if A = 1234 and B = 3412, applying the following two sequences of commands is changing the register content from A to B.

$1234 \rightarrow_L 2341 \rightarrow_L 3412$
$1234 \rightarrow_R 4123 \rightarrow_R 3412$

Therefore, your program should print LL or RR as an output for the above input.

Beware the cases where n contains 0 as a decimal digit. For instance, applying L to 1000 produces 1 since rotating 1000 to the left results in 0001. But, applying R to 1000 produces 100 since rotating 1000 to the right results in 0100.

Input

Your program is to read the input from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case consists of two integers, $\boldsymbol{A}$ and $\boldsymbol{B}$ ($\boldsymbol{A} \neq \boldsymbol{B}$), separated by a space, where the former denotes the initial content of the register and the latter denotes the wanted final content of it. Both $\boldsymbol{A}$ and $\boldsymbol{B}$ are greater than or equal to 0 and less than 10,000.

Output

Your program is to write to standard output. Print a minimal sequence of commands required to change the content of the register from A to B.

The following shows sample input and output for three test cases.

Sample Input	Output for the Sample Input
3 1234 3412 1000 1 1 16	LL L DDDD

Problem E
Goldbach's Conjecture

A natural number is called a prime number (or a prime) if it is bigger than 1 and has no divisors other than 1 and itself. For example, 5 is prime, since no number except 1 and 5 divides it. On the other hand, 6 is not a prime since $6 = 2 \times 3$.

Goldbach's conjecture is one of the famous unsolved problems in number theory and in all of mathematics. It states: Every even integer greater than 2 can be expressed as the sum of two primes. Such a number is called a Goldbach number. Expressing a given even number as a sum of two primes is called a Goldbach partition of the number. For example, $4 = 2 + 2$, $6 = 3 + 3$, $8 = 3 + 5$, $10 = 7 + 3$ or $10 = 5 + 5$, $12 = 5 + 7$, $14 = 3 + 11$ or $14 = 7 + 7$. Note that Goldbach partition has been found for any even interger n less than 10000.

Given any even integer n greater than 2, write a program that prints the two primes of the Goldbach partition of n. If there are more than one Goldbach partitions of n, find a partition such that the difference of the two primes of it is minimized.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case consists of an even integer n $(4 \le n \le 1{,}000)$.

Output

Your program is to write to standard output. For each test case, find the Goldbach partition as described above, and print its two primes in non-decreasing order with one blank between them.

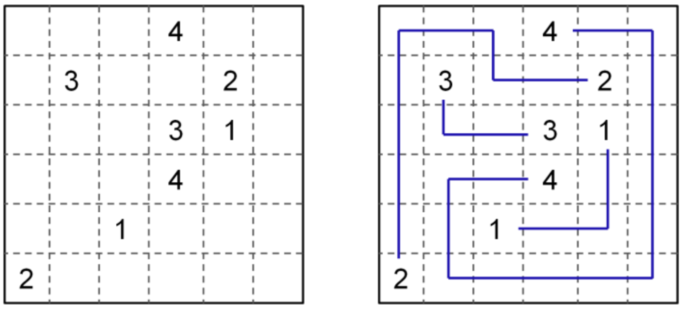
The following shows sample input and output for two test cases.

Sample Input	Output for the Sample Input
3 8 10 16	3 5 5 5 5 11

The 36th Annual ACM
International Collegiate Programming Contest
Asia Regional – Daejeon
Nationwide Internet Competition

Problem F
Number Link

Number Link is a type of logic puzzle involving finding paths to connect numbers in a grid. The player has to pair up all the matching numbers on the grid with continuous lines (or paths). The lines cannot branch off or cross over each other, and the numbers have to fall at the end of each line (i.e., not in the middle). Also the lines should pass by each cell in the grid exactly once. An example of Number Link with its solution is shown below.



(a) Initial configuration (b) Solution

Figure 1. An example of a Number Link Puzzle

Here we concentrate on the case that a pair of only one numbers, say number 1, is located on the grid. Also for the given $m \times n$ grid, m and n are both even. An example for this case is shown in the following Figure 2.

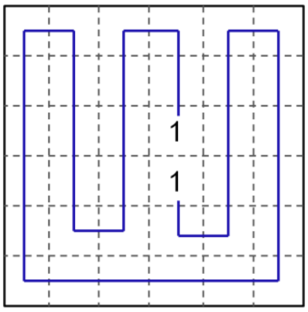


Figure 2. An example in which a pair of number 1's exists

Given an $m \times n$ grid where a pair of number 1's is located, write a program to find a path which connects the number 1's and passes by each cell exactly once.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. The first line of each test case contains two even integers $\boldsymbol{m}$ and $\boldsymbol{n}$, the dimensions of the $\boldsymbol{m} \times \boldsymbol{n}$ grid, where $\mathbf{2} \leq \boldsymbol{m}, \boldsymbol{n} \leq \mathbf{100}$. Each of the second and third lines of each test case contains two integers $\boldsymbol{i}$ and j, representing the location $(\boldsymbol{i}, \boldsymbol{j})$ of one of the number 1's in the grid. Here the location $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ of the number 1 denotes the cell of the $\boldsymbol{\alpha}$-th row and the $\boldsymbol{\beta}$-th column in the grid.

Output

Your program is to write to standard output. For each test case, if there is no path connecting the number 1's , then print -1. Otherwise, print 1, and in the next mn lines, print the two integers v and w, representing the (v, w) cell on the path. Here the end cells of the path are the locations of the number 1's, that is, print the locations of the number 1's in the first line and the last line, respectively.

The following shows sample input and output for two test cases.

Sample Input	Output for the Sample Input

<pre>2
4 4
2 2
3 3
6 6
3 4
4 4</pre>
	<pre>-1
	1
	4 4
	5 4
	5 5
	4 5
	3 5
	2 5
	1 5
	1 6
	2 6
	3 6
	4 6
	5 6
	6 6
	6 5
	6 4
	6 3
	6 2
	6 1
	5 1
	4 1
	3 1
	2 1
	1 1
	1 2
	2 2
	3 2
	4 2
	5 2
	5 3
	4 3
	3 3
	2 3
	1 3
	1 4
	2 4
	3 4</pre>

Problem G
Stains

Taeyeon has been using a rectangular carpet for a long time, thus the carpet has lots of indelible stains. Taeyeon decided to repair it by covering the stains with additional carpet patches. The shape of the patches is a diamond, which is a 45°- rotated axis-aligned square, as illustrated in Figure 1 below. The cost of a patch is proportional to its area. Taeyeon wants to minimize the total cost of the patches used to cover the stains, i.e., minimize the sum of area of the patches used. Moreover, Taeyeon is full of artistic sensitivity, so she wants some pattern of the patches. For this, she will align the centers of the patches on a line as illustrated in Figure 1. Patches can overlap each other.

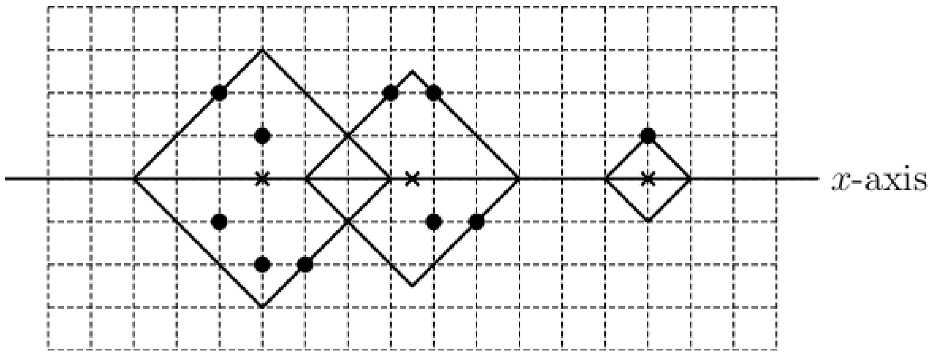


Figure 1. Stains are black dots and the centers of the patches are the crosses. The thick line is the x-axis.

The figure above shows an example in which the nine stains can be covered by three patches with the minimum cost of $32.5(= 18 + 12.5 + 2)$.

The stains are given with integer coordinates. You should find a set of diamond-shape patches such that (1) every stain must be contained in the interior or on the boundary of some patch, (2) the centers of the patches must be on x-axis, and (3) the sum of area of the patches used must be minimized. You can assume that stains are all distinct and no stains lie on the x-axis.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with integer $\boldsymbol{n}$, the number of stains, where $\mathbf{1} \leq \boldsymbol{n} \leq \mathbf{10,000}$. Each of the following $\boldsymbol{n}$ lines contains two integers, representing $\boldsymbol{x}$-coordinate and $\boldsymbol{y}$-coordinate of the stains between $-\mathbf{10^6}$ and $+\mathbf{10^6}$, inclusive.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain a real value, the minimum area of the patches used to cover the stains; you output only 1 digit after decimal point, simply ignore the ones from the 2nd digit after decimal point.

The following shows sample input and output for two test cases.

Sample Input	Output for the Sample Input
2 10 4 2 14 1 5 -2 5 1 8 2 4 -1 9 2 10 -1 6 -2 9 -1 9 2 1 6 3 11 3 16 6 19 2 20 1 21 3 21 2 24 3	32.5 146.0

Problem H
Training Season

Two professional baseball teams, X and Y, will have a training travel by visiting 7 cities = { c_1 , c_2 , c_3 , c_4 , c_5 , c_6 , c_7 } for a fixed period. X and Y start with the training season on a fixed day and also end on a day. So the starting day and ending day for this training travel should be same for two teams. For a training season, each team will have either training game on a city or take a rest in one hotel in a day. The visiting order of cities is determined in advance before starting training season. But each team can take a hotel rest by keeping the visiting order.

For example, the visiting order of a team is S=< c_1, c_2, c_3, c_1, c_4, c_1, c_5, c_4 > then each team can rearrange by taking a whole day rest in between visiting cities. Then the following schedules, S_1 and S_2 are also feasible where the visiting order of cities are unchanged. Note that 'R' denotes one resting day.

S_1=< c_1, c_2, R, c_3, c_1, R, R, c_4, c_1, c_5, R, c_4 >
S_2=< R, R, c_1, c_2, c_3, c_1, R, c_4, R, c_1, c_5, c_4 >

One day charge of the baseball stadiums is *C* for 7 cities. There is one way to save the rental charge for both X and Y teams. If two teams X and Y have a joint training together in a city stadium, then they can save the rental charge by paying *C/2* each. So X and Y hope to have a training game together to save rental charges by arranging the training days and resting days. But there is also expense for two teams to take a rest in hotels.

During training season, each team needs a rest sometimes to restore the physical strength. When a team takes a rest without training on a day, then that team should rent one hotel wholly. When a team occupies a hotel wholly, then the rental rate should be paid depending on the staying period. When a team reserves a hotel, then a team should pay *D* for the preemptive usage. And each team should pay *d* for each day. In case that a team reserves a hotel for *w* days consecutively, they the total rate should be $D+w*d$. For example if a team stays 5 consecutive days in a hotel where $D = 4$ and $d=1$, then they should pay $4 + 1*5 = 9$. And if a team stays 5 days separately, then the teams should pay preemptive rate *D* five times with 5∗d. So the total amount of hotel rate is $5 * 4 + 5 * 1 = 25$.

Assume that the order of visiting city is as follows, where 'R' denotes the one day hotel resting. Since the starting and ending day should be same, Y team should take a rest on Day7 and Day8 to make the same ending day of the team X.

Table 1. A Simple Schedule A

	Day1	Day 2	Day 3	Day 4	Day 5	Day 6	Day7	Day8
X	c_1	c_3	c_4	c_5	c_2	c_6	c_6	c_1
Y	c_3	c_4	c_2	c_6	c_6	c_1	*R*	*R*

When C=3 and D=4, d=1, then let us compute the total cost of training season trip.

Table 2. Expenses for Schedule A appeared in Table 1.

Day	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8
X	c_1	c_3	c_4	c_5	c_2	c_6	c_6	c_1
Y	c_3	c_4	c_2	c_6	c_6	c_1	*R*	*R*
X pays	*3*	*3*	*3*	*3*	*3*	*3*	*3*	*3*
Y pays	*3*	*3*	*3*	*3*	*3*	*3*	*5*	*1*
Total	*6*	*6*	*6*	*6*	*6*	*6*	*8*	*4*

So the total expense of two teams X and Y is 6*6 + 8 + 4 = 48. Now let us find another schedule for X and Y in order to have joint practice in the same city to save money. Here Table 3 shows another schedule.

Table 3. Schedule B

Day	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8
X	c_1	c_3	c_4	c_5	c_2	c_6	c_6	c_1
Y	*R*	c_3	c_4	*R*	c_2	c_6	c_6	c_1
X pays	*3*	*1.5*	*1.5*	*3*	*1.5*	*1.5*	*1.5*	*1.5*
Y pays	*5*	*1.5*	*1.5*	*5*	*1.5*	*1.5*	*1.5*	*1.5*
Total	*8*	*3*	*3*	*8*	*3*	*3*	*3*	*3*

Then since the total expense is 3∗6 + 8∗2 = 34, we can save 48-34 = 14 comparing with schedule A in Table 1. But note that the optimal schedule would be changed according to the value of *C*, *D* and *d*.

Your task is to compute the minimum expense for two teams X and Y by arranging resting days optimally into the sequence of visiting cities. So it is desirable to have a joint training in a city as many as possible we can achieve by considering the hotel rate for resting days depending on the value of *C* and *D*, *d* will be given differently in each test case. The sample input and the corresponding output will be helpful to understand these constraints and how to arrange the training schedule optimally.

Input

Your program is to read from standard input. The input consists of *T* test cases. The number of test cases *T* is given in the first line of the input. Each test case consists of three lines. The first line of each test case contains three integers to denote the value of C, D and d, respectively. C, D and d are all positive integers less than 10. Then the following two lines denote the order of visiting cities. Each city is denoted by one positive integer in {1,2,3,4,5,6,7}. The end of these two lines is marked by the number zero. The number of visiting cities, *N*, for each team is $2 < N < 100$.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain the minimum expenses for both teams X and Y by optimally arranging the visiting cities and resting days.

The following shows sample input and output for five test cases.

Sample Input	**Output for the Sample Input**
5 4 5 2 3 5 6 7 5 0 3 5 6 7 0 10 5 1 1 2 3 4 5 6 7 0 2 3 4 5 6 7 1 0 2 9 1 1 2 3 4 5 6 7 0	27 92 28 115 51

2 3 4 5 6 7 1 0 8 1 3 1 2 3 4 5 6 7 6 5 4 3 0 2 5 4 0 4 3 3 5 6 7 1 4 5 6 7 0 3 5 6 7 0	

Problem I
Two Numbers

Given a set of *distinct* integers $S = \{a_1, a_2, \ldots, a_n\}$, and a given target number K, your task is to find two different numbers of S which add up nearest to K.

For example, within a set of 10 integers

$$S = \{-7, 9, 2, -4, 12, 1, 5, -3, -2, 0\}$$

there is a pair of numbers {-4, 12} of S which add up to K = 8. Within the same set S, there are five pairs {{-7, 12},{9, -4}, {5, -2}, {5, 0}, {1, 2}} of numbers which add up nearest to K = 4.

Given a set of distinct integers and a target number, write a program to count the number of pairs of integers within the set which add up nearest to the target number.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case consists of two lines. The first line of each test case contains two integers. The first integer n $(2 \le n \le 1000000)$ is the number of integers in the set S, and the second integer K $(-10^8 \le K \le 10^8)$ is the target number. The second line of each test case contains n integers in the set S, where all integers are between -10^8 and 10^8, both inclusive.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain the number of pairs of integers in S which add up nearest to the target number K.

The following shows sample input and output for three test cases.

Sample Input	Output for the Sample Input
3 10 8 -7 9 2 -4 12 1 5 -3 -2 0 10 4 -7 9 2 -4 12 1 5 -3 -2 0 4 20 1 7 3 5	1 5 1

Problem J
Widest Path

We are given a graph which represents connections between nodes in the computer network, and the weight of an edge represents the bandwidth of a connection between two nodes. For the efficient data transmission between two nodes in the network, we are interested in finding a path between two nodes that has wide bandwidth. The *bandwidth of a path* is the minimum weight of an edge in the path. The widest path problem is to find the path between two nodes that has the maximum possible bandwidth.

For example, the widest path from node 1 to node 4 in Figure 1 has bandwidth 25, and passes through node 3 and node 2. The widest path from node 6 to node 3 has bandwidth 30, and passes through node 5.

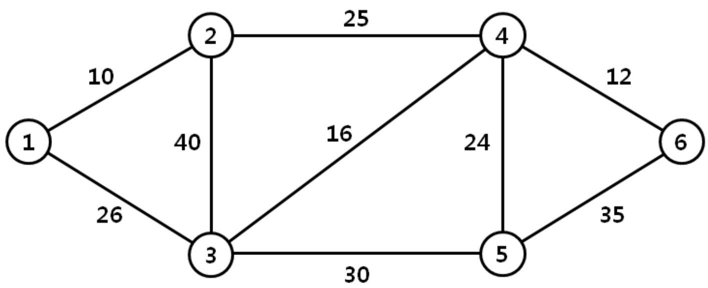


Figure 1. Example of a computer network

Given two nodes in a graph, write a program which determines the bandwidth of the widest path between two nodes.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with a line containing four integers n, m, s and t for a connected graph, where n $(2 \le n \le 1{,}000)$ represents the number of nodes and m $\left(1 \le m \le \frac{n(n-1)}{2}\right)$ represents the number of edges, s and t $(s \neq t)$ represents the two nodes(nodes are numbered from 1 to n). In the following m lines, the bandwidth of the edges are given; each line contains three integers, u, v, and b, where b $(1 \le b \le 10^5)$ is the bandwidth of a connection between two nodes u and v.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain the bandwidth of the widest path between two nodes s and t.

The following shows sample input and output for two test cases.

Sample Input	**Output for the Sample Input**
2 6 9 1 4 1 2 10 3 1 26 2 3 40 2 4 25 4 3 16 3 5 30 4 5 24 4 6 12 6 5 35 5 4 5 2 1 2 4 2 3 3 3 4 2 4 5 1	25 1